Slide 1

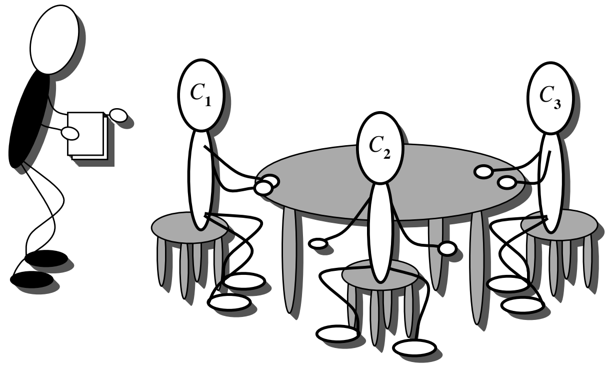
COMP3152/9152
Lecture 10
Applications to Security
Ron van der Meyden

Slide 2

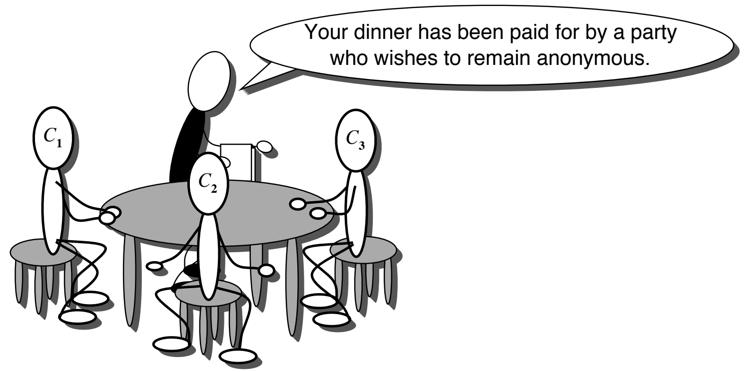
Chaum's Dining Cryptographers protocol

(Symbolic Model Checking the Knowledge of the Dining Cryptographers R. van der Meyden and K. Su, 17th IEEE Computer Security Foundations Workshop, Asilomar, June 2004, pp. 280-291)

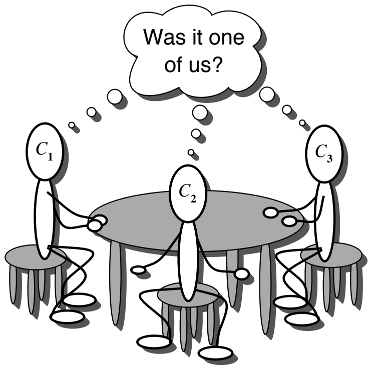
Slide 3



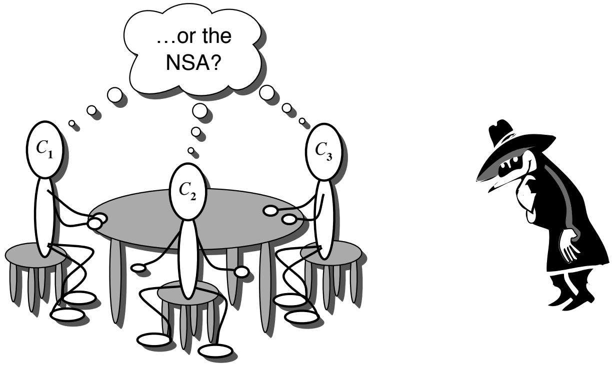
Slide 4



Slide 5



Slide 6



Slide 7

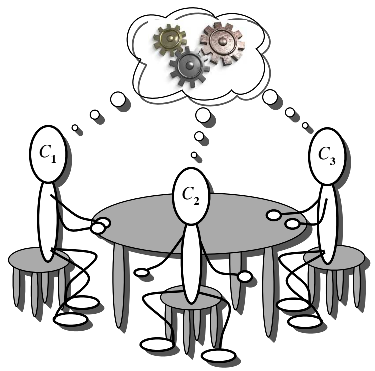
Knowledge Theoretic Specification

For Cryptographer C1:

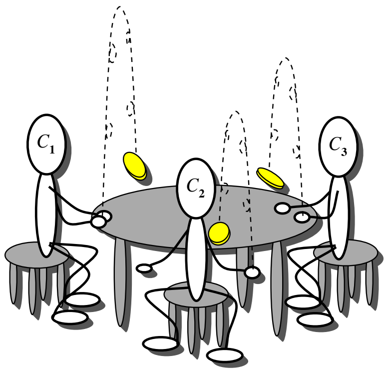
$$\begin{aligned}
&\neg paid(C1) \;\Rightarrow \\
&\quad Knows\ C1\ paid(NSA) \\
&\quad \vee\ [Knows\ C1\ (paid(C2) \vee paid(C3)) \wedge \\
&\qquad \neg\ Knows\ C1\ paid(C2)\ \wedge \\
&\qquad \neg\ Knows\ C1\ paid(C3)]
\end{aligned}$$

Similarly for the others ...

Slide 8

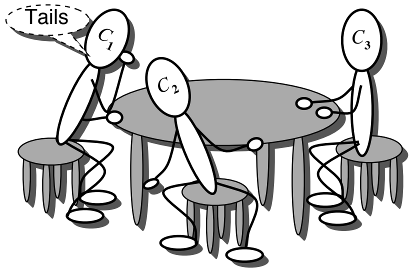


Slide 9



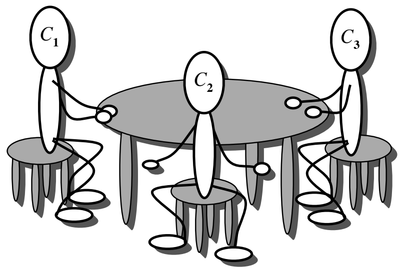
lide 10

1. Each C_i tells *only* his right neighbour what he tossed.



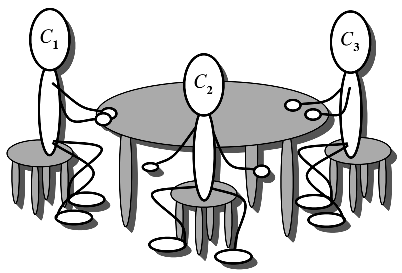
Slide 11

2. Each C_i announces whether the two coin tosses are equal – *unless* he paid.



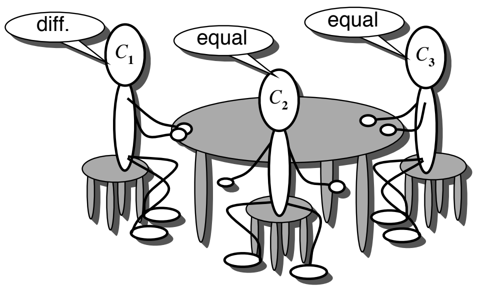
Slide 12

2. Each C_i announces whether the two coin tosses are equal – *unless* he paid.



lide 13

3. An *odd* number of “diff.” indicates one of the C_i paid.



lide 14

No. of Cryptographers:	3	4	5	10	20
fixed ordering	0.3	2.3	26.8	-	-
with sifting	.7	1.8	6.9	66	519

Table 1: Runtimes of Dining Cryptographers Verification (Seconds)

Slide 15

An observation

Given the protocol, the pattern of variable values observed by cryptographer 1 over time is very predictable:

– paid[1] is constant

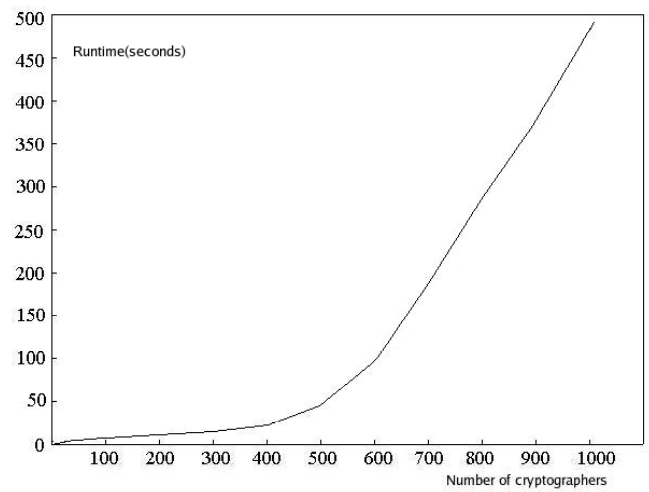
– said[i] changes only in the final step, for $i = 1, \ldots, n$

– coin[left] changes in step 2, then is constant

– coin[right] changes in step 3, then is constant

Upshot: we can reduce the representation of $o_0, \ldots, o_5$ from 5 copies of the above variables to 1.

Slide 16



lide 17

Another Security Protocol Example: Oblivious Transfer

Specification:

Alice has two messages $m_0, m_1 \in \{0,1\}^k$, unknown to Bob.

Bob selects whether he wants to receive m_0 or m_1.

Bob should learn only the message he selected.

Alice should not learn which message Bob selected.

lide 18

Rivest's solution, using an offline trusted third party

1. *Setup.* Ted chooses $r_0, r_1 \in \{0,1\}^k$ randomly and sends these values to Alice.
   Ted chooses $d \in \{0,1\}$ and sends d and r_d to Bob.
2. *Request.* Bob computes $e = c \oplus d$, where $\oplus$ denotes exclusive or, and sends it to Alice.
3. *Reply.* Alice computes $f_0 = m_0 \oplus r_e$ and $f_1 = m_1 \oplus r_{1-e}$ and sends f_0 and f_1 to Bob.
4. *Result.* Bob computes $m = f_c \oplus r_d$.

Slide 19

Intransitive Noninterference

What, indeed, is intransitive noninterference?, R. van der Meyden, Proc. European Symposium on Research in Computer Security, Dresden, Sept 2007, LNCS Vol. 4734, pp. 235-250.

Slide 20

Noninterference

Proposed by Goguen and Meseguer 1982

Context: Multi-level secure systems

partially ordered security levels $\Rightarrow$ transitive policies

Haigh and Young 87: extension to intransitive policies, deterministic systems

Rushby 1992: further results and corrections to Haigh and Young

van der Meyden 2007: improvement of Rushby theory

lide 21

Noninterference policies

Let D be a set of security domains.

A noninterference policy is a reflexive relation $\rightarrowtail \subseteq D \times D$

$u \rightarrowtail v$ means

> "actions of domain u are permitted to interfere with domain v", or
>
> "information is permitted to flow from domain u to domain v"

lide 22

Example

Public $\rightarrowtail$ Secret $\rightarrowtail$ Top-Secret

Public $\rightarrowtail$ Top-secret

but

Secret $\not\rightarrowtail$ Public, Top-Secret $\not\rightarrowtail$ Secret, Top-Secret $\not\rightarrowtail$ Public

Slide 23

Semantics for Transitive Policies

For each $u \in D$ define the function $\texttt{purge}_u : A^* \rightarrow A^*$ such that $\texttt{purge}_u(\alpha)$ is the subsequence of all actions a in α such that $\texttt{dom}(a) \rightarrowtail u$.

The system M is said to be *secure with respect to the policy* $\rightarrowtail$ when for all $\alpha \in A^*$ and domains $u \in D$, we have $\texttt{obs}_u(s_0 \cdot \alpha) = \texttt{obs}_u(s_0 \cdot \texttt{purge}_u(\alpha))$.

An equivalent formulation:

For all sequences $\alpha, \alpha' \in A^*$ such that $\texttt{purge}_u(\alpha) = \texttt{purge}_u(\alpha')$, we have $\texttt{obs}_u(s_0 \cdot \alpha) = \texttt{obs}_u(s_0 \cdot \alpha')$.

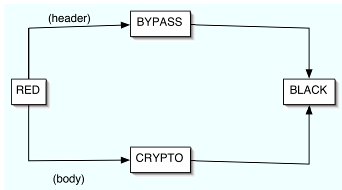
Slide 24

Motivation for Intransitive Policies

Downgrading:

$$H \rightarrowtail D \rightarrowtail L$$

Channel Control:



lide 25

Deterministic System Model

Machines of the form $\langle S, s_0, A, \rightarrow, \texttt{obs}, \texttt{dom}\rangle$ where

1. S is a set of states,
2. $s_0 \in S$ is the *initial state*,
3. A is a set of actions,
4. $\texttt{dom} : A \rightarrow D$ associates each action to an element of the set of security domains D,
5. $\rightarrow : S \times A \rightarrow S$ is a deterministic transition function, and
6. $\texttt{obs} : S \times D \rightarrow O$ associates an observation in some set O with each security domain.

Notation: $s \cdot \alpha$ for the state reached by performing the sequence of actions $\alpha \in Actions^*$ from state s.

lide 26

Haigh and Young's Semantics (1987)

Given a sequence of actions $a_1 \ldots a_n \in Actions^*$ and domain u , the *intransitive purge* $\texttt{ipurge}_u(\alpha)$ is the subsequence of all actions a_i such that there exists

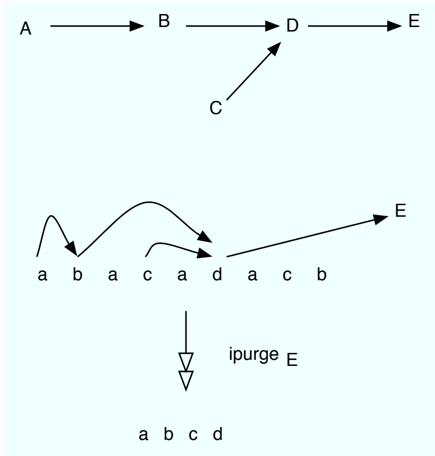
$$i = i_1 < i_2 < \ldots < i_k$$

with

$$\texttt{dom}(a_{i_1}) \rightarrowtail \texttt{dom}(a_{i_2}) \rightarrowtail \ldots \rightarrowtail \texttt{dom}(a_{i_k}) \rightarrowtail u$$

Slide 27

Example:



Slide 28

Haigh and Young's definition: IP-security

A system M is IP-secure with respect to a (possibly intransitive) policy $\rightarrowtail$ if for all $u \in D$ and all sequences $\alpha, \alpha' \in A^*$ with $\texttt{ipurge}_u(\alpha) = \texttt{ipurge}_u(\alpha')$, we have $\texttt{obs}_u(s_0 \cdot \alpha) = \texttt{obs}_u(s_0 \cdot \alpha')$.

lide 29

(Perfect Recall) Knowledge in Asynchronous Systems

Define the view of domain u with respect to a sequence $\alpha \in A^*$ to be the sequence of all observations of u and actions *of* u while running α, with stuttering observations reduced to a single occurrence:

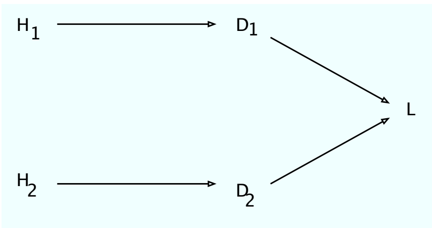
E.g., if running $\alpha = hhlh$ produces $o_1 o_1 o_1 l o_2 o_2$ at L then then $\mathtt{view}_L(\alpha) = o_1 l o_2$

define $\sim_u$ on sequences of actions by $\alpha \sim_u \alpha'$ if $\mathtt{view}_u(\alpha) = \mathtt{view}_u(\alpha)$.

$\alpha \models K_u \phi$ if $\alpha' \models \phi$ for all $\alpha' \sim_u \alpha$

lide 30

Example



Define the system M with

1. actions: h_1, h_2, d_1, d_2, l of domains H_1, H_2, D_1, D_2, L respectively.
2. states: the set of all strings in A^*.
3. transitions: $\rightarrow (\alpha, a) = \alpha a$ for $\alpha \in A^*$ and $a \in A$,
4. $\mathtt{obs}_u(\alpha) = [\mathtt{ipurge}_u(\alpha)]$.

Slide 31

Let $\alpha_1 = h_1 h_2 d_1 d_2$

Then $\mathtt{obs}_L(\alpha_1) = [\mathtt{ipurge}_L(\alpha_1)] = [\alpha_1]$

Let p="there was an h_1 before an h_2"

p is a fact about H_1, H_2.

$\alpha_1 \models K_L p$

Slide 32

But

$$\begin{aligned} & \mathtt{view}_{D_1}(\alpha_1) \\ & = \mathtt{view}_{D_1}(h_1 h_2 d_1 d_2) \\ & = [\epsilon] \circ [h_1] \circ [h_1] \circ d_1 \circ [h_1 d_1] \circ [h_1 d_1] \\ & = [\epsilon] \circ [\epsilon] \circ [h_1] \circ d_1 \circ [h_1 d_1] \circ [h_1 d_1] \\ & = \mathtt{view}_{D_1}(h_2 h_1 d_1 d_2) \end{aligned}$$

Similarly, $\mathtt{view}_{D_2}(\alpha_1) = \mathtt{view}_{D_2}(h_2 h_1 d_1 d_2)$

So

$$\alpha_1 \models K_L p \wedge \neg D_{\{D_1, D_2\}} p$$

lide 33

An alternative definition - TA security

Given a policy $\rightarrowtail$, define, for each agent $u \in D$, the function $\texttt{ta}_u$, with domain A^*, inductively by $\texttt{ta}_u(\epsilon) = \epsilon$, and, for $\alpha \in A^*$ and $a \in A$,

$$\texttt{ta}_u(\alpha a) = \begin{cases} \texttt{ta}_u(\alpha) & \text{if } \texttt{dom}(a) \not\rightarrowtail u \\ (\texttt{ta}_u(\alpha), \texttt{ta}_{\texttt{dom}(a)}(\alpha), a) & \text{if } \texttt{dom}(a) \rightarrowtail u \end{cases}$$

Define a system M to be TA-secure with respect to a policy $\rightarrowtail$ if for all agents u and all $\alpha, \alpha' \in A^*$ such that $\texttt{ta}_u(\alpha) = \texttt{ta}_u(\alpha')$, we have $\texttt{obs}_u(s_0 \cdot \alpha) = \texttt{obs}_u(s_0 \cdot \alpha')$.

lide 34

How these definitions are related

Theorem 1

1. *P-secure* $\Rightarrow$ *TA-secure* $\Rightarrow$ *IP-secure.*
2. *If* $\rightarrowtail$ *is transitive then P-secure = TA-secure = IP-secure.*

Slide 35

Unwinding and Access Control Models

Slide 36

Access Control

A *system with structured state* is a machine $\langle S, s_0, A, \rightarrow, \texttt{obs}, \texttt{dom}\rangle$ together with

1. a set N of *names*,
2. a set V of *values*, and functions
3. $\texttt{contents} : S \times N \rightarrow V$, with $\texttt{contents}(s, n)$ interpreted as the value of object n in state s,
4. $\texttt{observe} : D \rightarrow \mathcal{P}(N)$, with $\texttt{observe}(u)$ interpreted as the set of objects that domain u can observe, and
5. $\texttt{alter} : D \rightarrow \mathcal{P}(N)$, with $\texttt{alter}(u)$ interpreted as the set of objects whose values domain u is permitted to alter.

Slide 37

For a system with structured state, when $u \in D$ and s is a state, define $\texttt{state}_u(s) : \texttt{observe}(u) \rightarrow V$ by $\texttt{state}_u(s)(n) = \texttt{contents}(s, n)$ for $n \in \texttt{observe}(u)$.

Define a binary relation $\sim_u^{oc}$ of *observable content equivalence* on S for each domain $u \in D$, by $s \sim_u^{oc} t$ if $\texttt{state}_u(s) = \texttt{state}_u(t)$.

Slide 38

Rushby's Reference Monitor Assumptions

RM1. If $s \sim_u^{oc} t$ then $\texttt{obs}_u(s) = \texttt{obs}_u(t)$.

RM2. If $s \sim_{dom(a)}^{oc} t$ and either $\texttt{contents}(s \cdot a, n) \neq \texttt{contents}(s, n)$ or $\texttt{contents}(t \cdot a, n) \neq \texttt{contents}(t, n)$ then $\texttt{contents}(s \cdot a, n) = \texttt{contents}(t \cdot a, n)$

RM3. If $\texttt{contents}(s \cdot a, n) \neq \texttt{contents}(s, n)$ then $n \in \texttt{alter}(dom(a))$.

RM2 is equivalent to the following: For all states s, either

1. for all $t \sim_{dom(a)}^{oc} s$, we have $\texttt{contents}(t \cdot a, n) = \texttt{contents}(t, n)$, or
2. for all $t \sim_{dom(a)}^{oc} s$, we have $\texttt{contents}(s \cdot a, n) = \texttt{contents}(t \cdot a, n)$

Slide 39

Consistency of an access control system with a policy:

AOI. If $\texttt{alter}(u) \cap \texttt{observe}(v) \neq \emptyset$ then $u \rightarrowtail v$.

Proposition 1 *(Rushby 92) Suppose M is a system with structured state that satisfies RM1-RM3 and AOI. Then M is IP-secure for $\rightarrowtail$.*

Slide 40

A weaker notion of Access Control

[RM2′] For all actions a states s, t and names $n \in \texttt{alter}(dom(a))$, if $s \sim_{\texttt{dom}(a)}^{oc} t$ and $\texttt{contents}(s, n) = \texttt{contents}(t, n)$ we have $\texttt{contents}(s \cdot a, n) = \texttt{contents}(t \cdot a, n)$.

Example: n is a block of memory, a writes to a single location

Say M a system with structured states is a *weak access control system* compatible with $\rightarrowtail$ if it satisfies RM1, RM2′, RM3 and AOI.

Slide 41

Proposition 2 *If M is a* weak *access control system compatible with $\rightarrowtail$ then M is TA-secure (hence IP-secure) for $\rightarrowtail$.*

Slide 42

Unwinding Conditions

Suppose we have for each domain u an equivalence relation $\sim_u$ on the states of M.

OC: If $s \sim_u t$ then $O_u(s) = O_u(t)$. (Output Consistency)

SC: If $s \sim_u t$ then $s \cdot a \sim_u t \cdot a$. (Step Consistency)

LR: If not $\mathtt{dom}(a) \rightarrowtail u$ then $s \sim_u s \cdot a$. (Left Respect)

If these conditions are satisfied then M is secure with respect to a transitive policy (Goguen & Meseguer 84).

Slide 43

Completeness of Unwinding (Transitive Policies)

Proposition 3 *(Rushby 92) Suppose M is P-secure with respect to the transitive policy $\rightarrowtail$. Then there exist equivalence relations $\sim_u$ on the states of M with respect to which M satisfies OC, SC and LR.*

(Specifically, $s \approx_u t$ if for all strings α in A^* we have $O_u(s \cdot \alpha) = O_u(t \cdot \alpha)$.)

Slide 44

Unwinding Intransitive Noninterference

WSC: If $s \sim_u t$ and $s \sim_{dom(a)} t$ then $s \cdot a \sim_u t \cdot a$.
(Weak Step Consistency)

Proposition 4 *(Rushby 92) Suppose that $\sim_u$ are equivalence relations on the states of a system M that satisfy OC,WSC and LR. Then M is IP-secure for $\rightarrowtail$.*

(But no completeness result.)

lide 45

Unfolding a system

Given a system $M = \langle S, s_0, \rightarrow, \texttt{obs}, \texttt{dom}\rangle$ with actions A, define the system $\texttt{uf}(M) = \langle S', s'_0, \rightarrow', \texttt{obs}', \texttt{dom}\rangle$ with actions A by

1. $S' = A^*$
2. $s'_0 = \epsilon$
3. $\rightarrow' (\alpha, a) = \alpha a$, for $\alpha \in S'$ and $a \in A$
4. $\texttt{obs}'_u(\alpha) = \texttt{obs}_u(s_0 \cdot \alpha)$ (RHS in M)

$\texttt{uf}(M)$ is *bisimilar* to M (in the expected sense)

lide 46

Say that a system M with states S *admits a weak access control interpretation compatible with* $\rightarrowtail$ if there exists

1. a set of names N
2. a set of values V and functions
3. $\texttt{observe} : D \times S \rightarrow \mathcal{P}(N)$,
4. $\texttt{alter} : D \times S \rightarrow \mathcal{P}(N)$ and
5. $\texttt{contents} : N \times S \rightarrow V$

with respect to which M is a weak access control system compatible with $\rightarrowtail$.

Slide 47

Theorem 2 *The following are equivalent*

1. *M is TA-secure with respect to $\rightarrowtail$*
2. *$\texttt{uf}(M)$ admits a weak access control interpretation compatible with $\rightarrowtail$;*
3. *there exist equivalence relations $\sim_u$ on the states of $\texttt{uf}(M)$ satisfying OC,WSC and LR;*

(So, weak unwinding incomplete for IP-security on two counts: unwinding is complete for the stronger TA-security, wrt $\texttt{uf}(M)$ rather than M).